Available online at www.sciencedirect.com

ScienceDirect

Procedia Engineering 64 (2013) 104 – 114

Procedia Engineering

www.elsevier.com/locate/procedia

International Conference On DESIGN AND MANUFACTURING, IConDM 2013

SDR based Multi Data Communication System Design

Mr.Praveen Kumar P and Dr.Noor Mahammad Sk*

*Master of Design student and *Asst.Professor, IIITD&M kancheepuram,chennai-600127,india*

Abstract:

This work proposes a way of transmitting different data like text, voice, image from a single transmitter to multiple receivers which are placed at different locations of campus. This will serve for the purpose of conveying campus related announcements. We are making use of unlicensed frequency band to achieve our purpose on board level design. The concept of software defined radio is used for transmitting the various data from single transmitter. The advantage of such implementation is same transmitter hardware can be used for future advances in communications.

Key words: Software defined radio; transmitter ; receiver; multi data communication; FPGA.

1. Introduction

Wireless networks works based on OSI layer model, as shown in Figure 1. In wireless network design, the physical and data link layers ensure a reasonably reliable point-to-point communication abstraction. New technologies are making all features of the radio can be defined by software and can configuration as a bit stream of the Tx/Rx and can reside on ROM and when ever required, we can configure on the FPGA chip as a Tx (Rx based on selection) by using physical layer in the OSI model. In the future there will be no need to replace your transceiver to upgrade it, it will merely require a download of a new software package to acquire the latest features. It should be possible to incorporate all ‘data’ modes (voice, text, and image) within the same package, making it a truly all mode transceiver. At present time, all SDR products should be viewed as experimental.

* Corresponding author. Tel.: +91-44-27476349; fax: +91-44-27476301.
E-mail address: noor@iiitdm.ac.in

doi:10.1016/j.proeng.2013.09.081

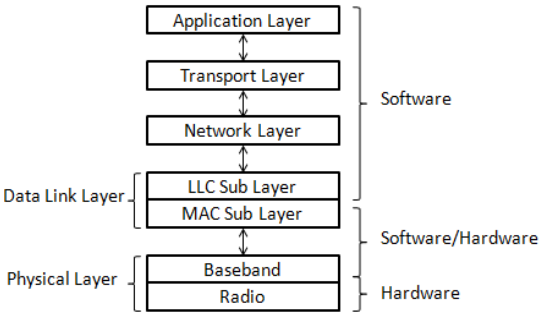

Fig.1 Overview of a wireless protocol stack

Wireless radio is very different from a wired radio in two major aspects 1st it is a shared medium, so any two concurrent communications in the same frequency band will interfere with each other 2nd is radio signal received by a node will suffer with various physical phenomena like noise, multipath fading and Doppler shift. Therefore, an efficient wireless protocol must adapt promptly to such changing channel characteristics. A protocol that satisfies this requirement is considered to be a cross-layer wireless protocol. Over the past few years, researchers have proposed many cross-layer protocols. Some of the examples are interference cancellation [1], ZigZag decoding [2], Conict Maps (CMAP) [3], the SoftPHY interface [4], Sample Width [5], Analog Network Coding [6], MIXIT [7], COPE [8], VWID [9], ODS [10], SWIFT [11] and MIM [12]. All these proposals advocate some kind of modifications to the current layering structure by exposing various channel information obtained from the physical layer to higher layers as shown in Figure 1.

Researchers have built radio platforms using Field Programmable Gate Array (FPGA) [13]. A FPGA chip contains programmable logic components that can be configured dynamically to execute a required hardware design, which is usually described in Hardware Description Language (HDL). Although a design running on FPGA is typically slower than its Application-Specific Integrated Circuit (ASIC) counterpart.

The organization of the paper is as follows 1st section explains about the introduction about the topic, 2nd section talks about our objective and transmitter & receiver Architecture designs, 3rd section describes about Interfaces between different hardware's to support multidata transmission, 4th section presents the data synchronization and data control using modulation schemes, 5th section is about the modulator design and its implementation on the FPGA, and the last section shows the implementation of modulation schemes, which will support multi data communication on FPGA through hardware-software co-simulation.

2. Design Objective

To design a communication device for campus applications, The transmitter and receiver will work based on software defined radio (SDR) principle. The design consists of RF circuit for transmission to cater the text, image and voice data. Core communication circuit implementation will be made on the FPGA. A standard digital interface and RF circuits are considered for our design. Receiver is designed to receive the text, image and voice data from a single transmitter. Digital interface between receiver and text display (LED matrix) units, image processing (JPEG - CODECS) hardware and its corresponding display (LCD screen) units, and voice processing hardware and speaker are designed.

2. 1 Transmitter Pipeline

TX Controller receives information from the MAC and generates the control and data for all the subsequent blocks, then Scrambler unit randomized data bit streams to remove redundancy present in data. This gives better results for Forward Error Correction (FEC). A scrambler is implemented with linear feedback shift registers (LFSR). Then FEC encoder block encodes data and adds repeated patterns to the bit stream to enable the receiver to detect and correct errors. FEC takes the help of Convolution encoder encoding before the data is passed to the convolution encoder. RS encoding itself has several parameters, for example, a (N=255, K=239, T=8) encoder takes 239 bytes of data and adds 2× 8 bytes of parity to produce a 255 byte coded message.

To make efficient transfer of bits over transmitter we use a block called puncturing, it will reduce no. of bits to be transmitted over Tx. Then to protect against burst errors Interleaver rearranges blocks of data bits by mapping adjacent coded bits into non-adjacent subcarriers. All protocols support BPSK, QPSK, 16-QAM and 64-QAM modulation schemes. Then the Map per block passes interleaved data through a serial to parallel converter, mapping groups of bits to separate carriers, and encoding each bit group by frequency, amplitude, and phase. Pilot/guard insertion adds the values for pilot and guard subcarriers. The subcarrier indices are protocol specific. Both protocols use scramblers to generate values for the pilots and use null values for the guard subcarriers. IFFT converts symbols from the frequency domain to the time domain. CP insertion copies some samples from the end of the symbol to the front to add some redundancy to the symbols. These duplicated samples are known as a cyclic prefix (CP). The aim of the cyclic prefix is to avoid Inter-Symbol Interference (ISI) caused by multipath propagation. This block also adds a preamble before the first symbol is being transmitted. A preamble is a collection of predefined complex numbers known by the receiver so that it can detect the start of new transmission. The preambles for the two protocols have similar structure. After CP insertion, the symbol are converted into analog signals by D/A converter and transmitted in the air. The transmitter architecture is shown in Figure 2

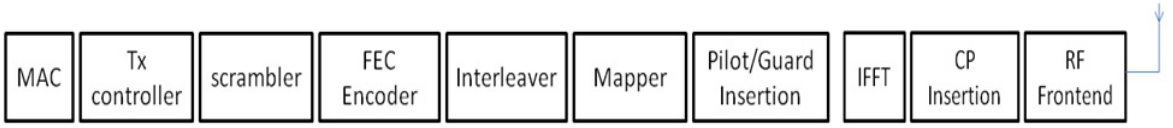

Fig. 2 Transmitter architecture

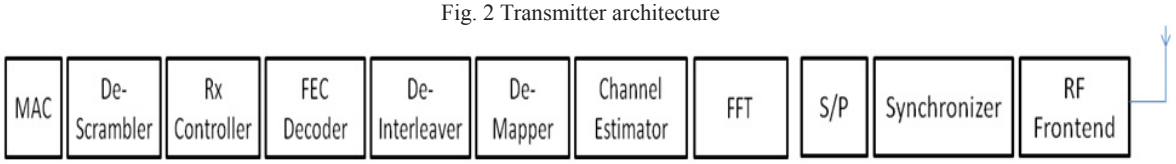

Fig. 3 Receiver Architecture

2.2 Receiver Pipeline

Receiver architecture is shown in Figure 3. The functionality of the blocks in the receiver is reverse of the functionality of their corresponding blocks in the transmitter. However, since the receiver has to recover data from a degraded signal, some receiver blocks have to do more processing on signals and it require more implementation effort. When the antenna detects the signal, it amplifies the signal and passes it to the A/D converter to generate baseband digital samples. Then Synchronizer detects the starting position of an incoming packet based on preambles. It is extremely important for the synchronizer to correctly estimate the OFDM symbol boundaries so that subsequent blocks process collection of samples together. In many implementations, the synchronizer also detects and corrects carrier frequency offset that is caused by the difference in the oscillator frequencies at transmitter and receiver or due to the Doppler Effect. The synchronizer uses the preamble to perform timing and frequency synchronization. There are many different implementations of the synchronizer, most of which involve auto-correlation and cross-correlation. For the synchronizer to support different protocols, it needs to know the preamble structure, the symbol size and the CP size of the protocol. Serial to Parallel (S/P) removes the cyclic prefix (CP) and then aggregates samples into symbols before passing them to the FFT. It also propagates the control information from the RX Controller in order to configure subsequent blocks. FFT converts OFDM symbols from the time domain back into the frequency domain. De-mapper demodulates data and converts samples to encoded bits, which are used by the FEC decoder. The number of encoded bits generated per sample is determined by the specific modulation scheme. The parameters of this block are modulation schemes supported and the functions for converting samples to decisions. De-interleaver reverses the interleaving performed by transmitter and restores the original arrangement of bits. FEC decoder uses the redundant information that was introduced at the transmitter to detect and correct errors that may have occurred during transmission. The Viterbi algorithm [14] is used to decode convolutionally encoded data. To support multiple protocols, the decoder uses the same parameter settings as the convolution encoder at the transmitter side. Descrambler reverses the scrambling performed by the transmitter. Receiver controller is designed based on the decoded data received from Descrambler, the RX Controller generates the control feedback to S/P block.

3. Radio Device Interface

We are feeding a base-band signal containing frequencies from DC to MHz range into FPGA through ADC as shown in Figure 6 from receiver. Computers can only understand digital signals comprising '0's and '1's. To send and receive signals on-air, the baseband signal must communicate with external devices like DACs, ADCs, and RF stage, signal from the antenna will be down converted into audio signal frequency and is given to FPGA. RF interface considered is soundcard, similarly signal from the interface will be up converted to RF stage at transmitter side, soundcard is the interface at input as well as output side with FPGA have been shown in Figure 4 and 5 below depicts and gives details about the sound card at Tx and Tx side are shown in Figure 6 and 7 respectively.

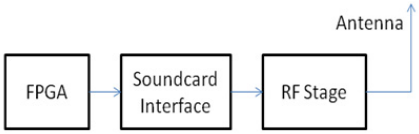

Fig.4 Prototype of the multi data transmitter

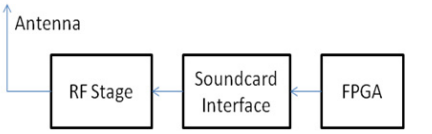

Fig.5 Prototype of the multi data receiver

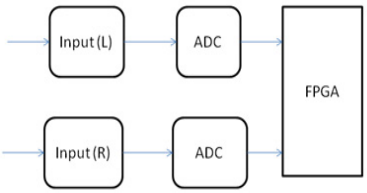

Fig .6 Sound Card (Input)

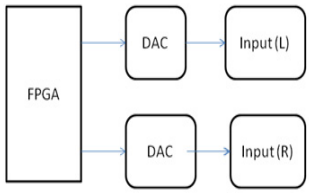

Fig .7 Sound Card (Output)

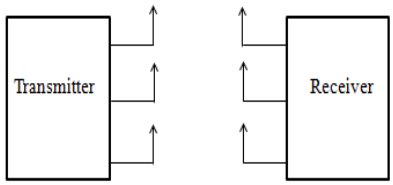

Fig. 8 switching of antenna

3.1 Identity the modulation switching scheme at Tx as well as Rx

In FPGA based communication we have various peripherals connected to the Spartan-3E board, which support Multi data communication, and to support as well as to control different data types like voice, text and image we have different modulation schemes, so we have to select the modulation schemes depending on the data being sent at the Tx side and receiver side, the prototype of the communication for our purpose have been shown in Figure 8, And we have considered two modulation schemes such as QPSK and BPSK to support our applications. The high speed switching membrane will do the job of transmitting and receiving based on application like voice, image, and text.

4. Synchronizing Data and Control

Modulation scheme is used as a data controlling scheme to achieve our purpose, Cross-layer protocol stacks require new ways of reconfiguring the lower layers (PHY layer, Data link layer) at run-time, and control paths are embedded in the design. The commands from the higher layer to enable reconfiguration are usually referred to as control to distinguish them from the actual data through the Tx pipeline. Consider the mapper module shown in Figure 9(a), which takes a data input (a group of bits to map into a PHY symbol) and a control input (the modulation that determines the mapping). For the mapper to function correctly, the modulation control should arrive at the same time as the data bits it applied. Sometimes the reconfiguration affects several modules and has to affect them in a coordinated way. For example, the MAC protocol requires rapid switching from receive to transmit, where the ongoing reception must be aborted and all modules must prepare to transmit. For correct operation, transmit data should be sent along the Tx pipeline only when all the modules have finished processing the control signal to abort and flush (Figure 9(b)). One can control and synchronize the various data types at Tx and Rx side by using modulation scheme as a control signal as a data-driven control, messages between blocks contain both control information and the set of data values the control must operate on. Control tokens are embedded into the data path along with the data, and are not modified by blocks that do not need to act on them as the message flows through the pipeline. The control information is stripped of when the message leave the pipeline

(a) (b)

Fig. 9: The problem of synchronization between control and data when reconfiguring a lower layer.

(a) Input data bits and corresponding modulation must arrive together at the mapper, which modulates data into PHY symbols.
(b) When aborting the reception of a packet at the PHY, new data should not be sent into the pipeline the pipeline until reconfig. Is complete at all the modules.

(a) (b)

Fig. 10: Examples illustrating data-driven control.
(a) The modulation control and data bits arrive together along the same input at the mapper.
(b) Aborting a reception is accomplished by forwarding an abort Token along the data path and withholding the data after the token until reconfiguration is complete and a response is received.

Control tokens can be interspersed with data at any granularity, enabling us to pass control with each bit or groups of bits, or per-packet. This approach incurs the overhead of extra hardware circuitry to pass and identify control tokens through the data path, but allows protocol designers to modify the structure of pipelines or refine any individual block easily without worrying about retiming the controls. Figure 10(b) shows an implementation of the examples in Figure 10(a) with data-driven control.

The concept of data-driven control is neither new to hardware systems nor to software systems. For example, packet transmissions in wormhole networks use header its (controls) to reserve buffers of each node along the paths until all the following body-its (data) pass through that node. Another example is that Click [15] uses packet annotations" to couple control and data together. The notion is also used in SDR-based systems [16].

5. Modulator Design

In accordance with [17], the modulator consists of two general parts, as shown in Figure 11. There is a baseband modulation signal processing block (Symbol mapper) and a carrier generation block (DDS synthesizer) with multipliers and adder. Symbol mapper switches the modulation type (BPSK, QPSK). In QPSK case every couple of bits is divided into two channels I and Q. The odd bit goes to the I channel and the even bit goes to the Q channel. Otherwise in BPSK case the whole dataflow goes to I channel and Q channel is unplugged. Using Saprtan-3E, BPSK modulated signal can be created. Proposed block diagram for BPSK implemented in FPGA is shown in Figure 12. Using ADC, message signal is converted to digital signal. There by, in FPGA, this signal can be processed and, FPGA based BPSK modem can be created. Bit separator block separates one by one in 14 bit output of ADC. Bit separator was created using VHDL Language in FPGA.

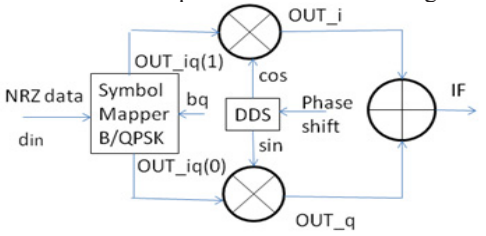

Fig. 11: Block diagram of modulator.

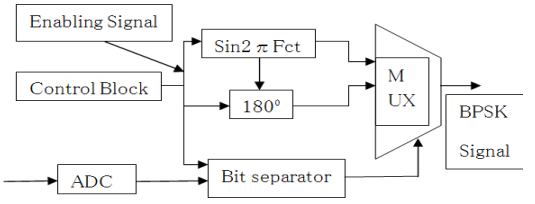

Fig 12: Principle of the BPSK system implemented on the FPGA

5.2 Modulation Scheme for text and voice for TX and Rx

The diagram shown below in Figure 13 is explains about how the QPSK modulator can be implemented on FPGA and how the data is being accessed from ROM (for reference) and can configure on FPGA as a bit stream.

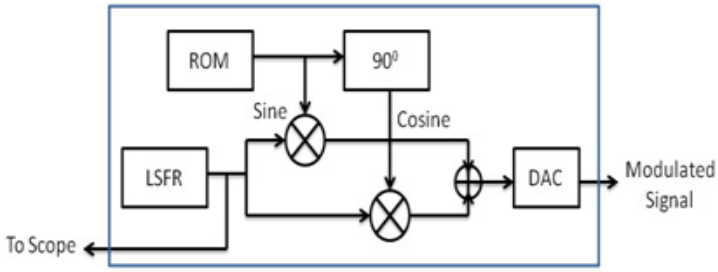

Fig. 13 The principle of the QPSK modulator on the FPGA.

5.3 Hardware and Software Co simulation of a QPSK modulation scheme

Among all applications QPSK modulation scheme will supports Data and voice so which has been realized using system generator software and implemented it on FPGA by using Spartan-3E board as shown in Figure 14. The results of QPSK are being analysed for further performance enhancement.

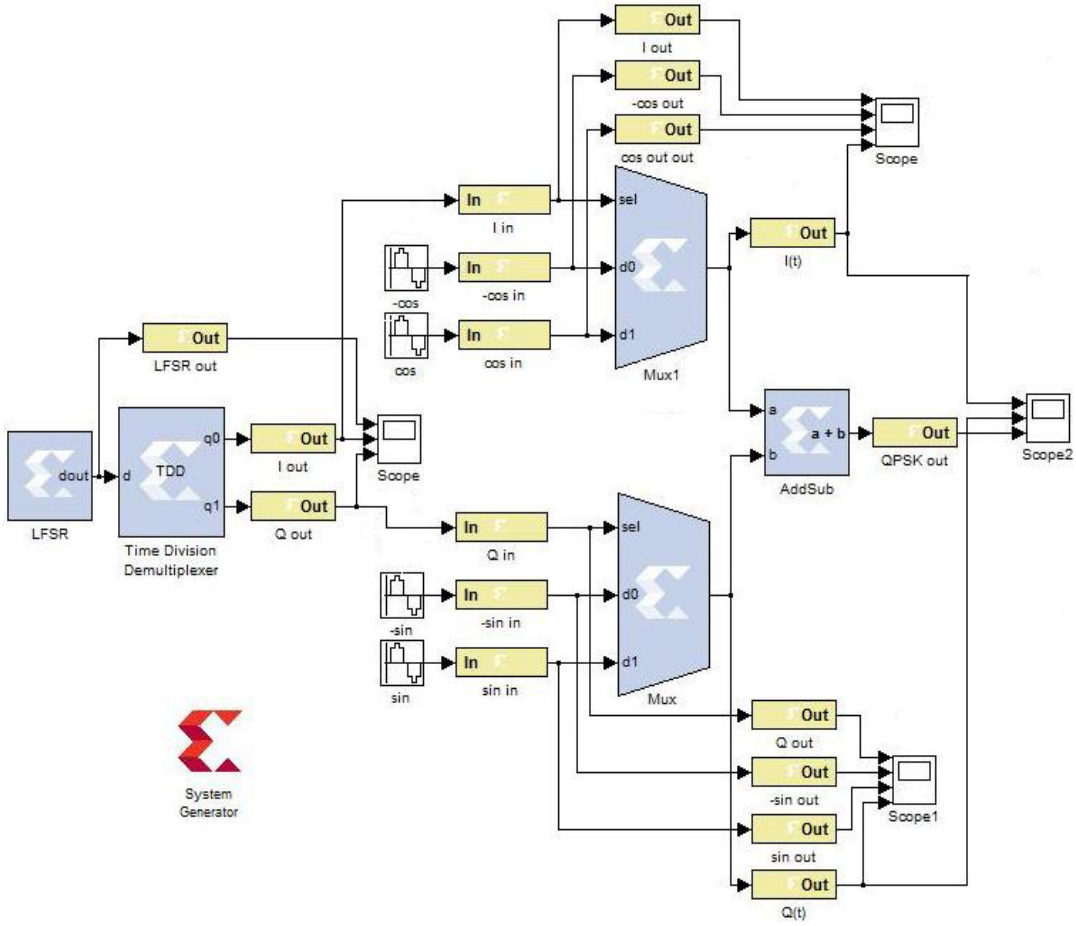

Fig.14 QPSK Modulator in System Generator.

6. Results and Analysis

6.1 System Simulation Result

The simulation model for punctured convolution encoder of rate 2/3 with constraint length 7 is used is shown in Figure 15. The data source (random integer generator) output is input to the "Gateway In" block. This block converts the data in double precision to the Xilinx fixed point representation. The output from the data output port 1 of the encoder is serial-to-parallel converted and given to the puncture block with puncture code 10. Every second, encoded output bit from data output port 1 is deleted by the puncture block after serial-to-parallel conversion. However, there is no puncture block on data output port 2 since the puncture code is 11 and all the encoded bits from output port 2 are transmitted. A constant value of 1 is used as input to the input port (vin) to specify to the encoder that the data on its input port is valid and is ready to be encoded. When there is valid output on the output ports of the encoder, the valid output port (Vout) output is set to high, which is sent to the Viterbi decoder valid input port.

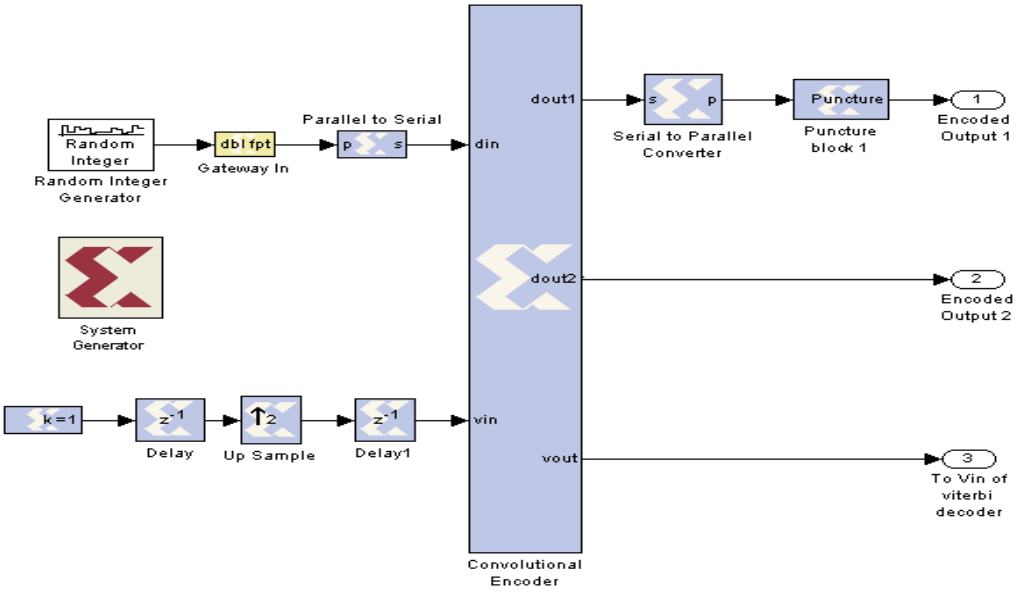

Fig.15 Encoding and puncturing

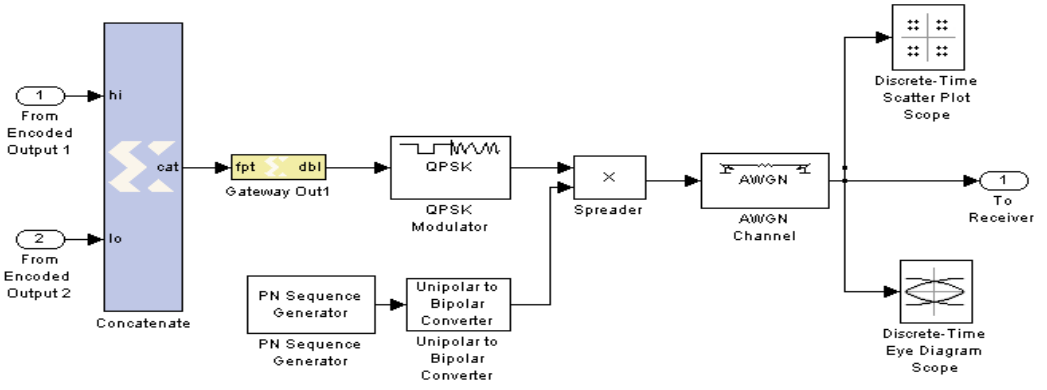

Fig.16 Modulation and spreading of the encoded signal

Figure 16 shows the modulation and spreading design process. The punctured data from the two puncture blocks are concatenated and then modulated using QPSK modulator. The data from the concatenation passes through a ”Gateway out” block so that the Xilinx’s fixed point representation is converted to SIMULINK double precision since QPSK modulation and spreading is done with SIMULINK blocks. After modulation, the modulated data is then spread using PN sequence generator and passed through the AWGN channel. At the receiver, the signal is despreaded using the same PN sequence generator and demodulated using QPSK demodulator. This is shown in Figure 17. After demodulation, the input data to the two depuncture blocks is first passed through the ”Gateway In” block converting it to Xilinx’s fixed point representation. Resulting data is then passed through the slicer to separate the encoded bits.

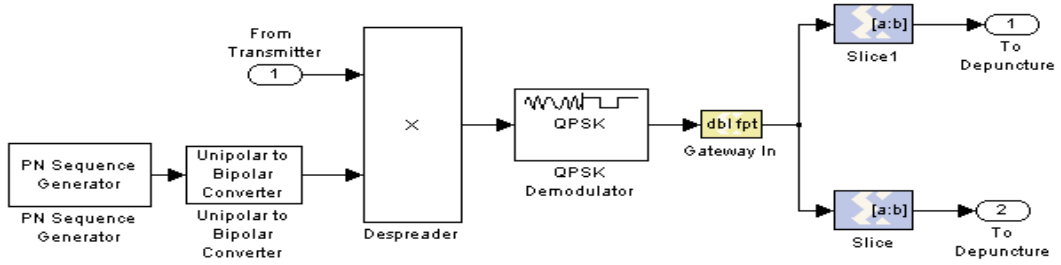

Fig.17 Dispreading and demodulation of the received signal.

Recall that the purpose of depuncturing and decoding is for forward error correction. The same puncture code 10 and 11 used for puncturing is used for depuncturing. Figure 18 shows the depuncturing and decoding blocks used in the simulation. The null-symbol is inserted after every input to the data input port 1 of the Viterbi decoder through the depuncture block and no null-symbol is inserted at the data input port 2 of the decoder as no bits were punctured during transmission. After depuncturing from block 1, the data is parallel-to-serial converted and input to the Viterbi decoder. The input to the second port of the Viterbi decoder and the input to the valid input port are given after few delays in order to synchronize with the output from depuncture block 1. At the decoder, the depunctured data is then decoded using a rate 1/2 Viterbi decoder. The decoder output is serial-to-parallel converted accordingly and the input and output waveform is compared.

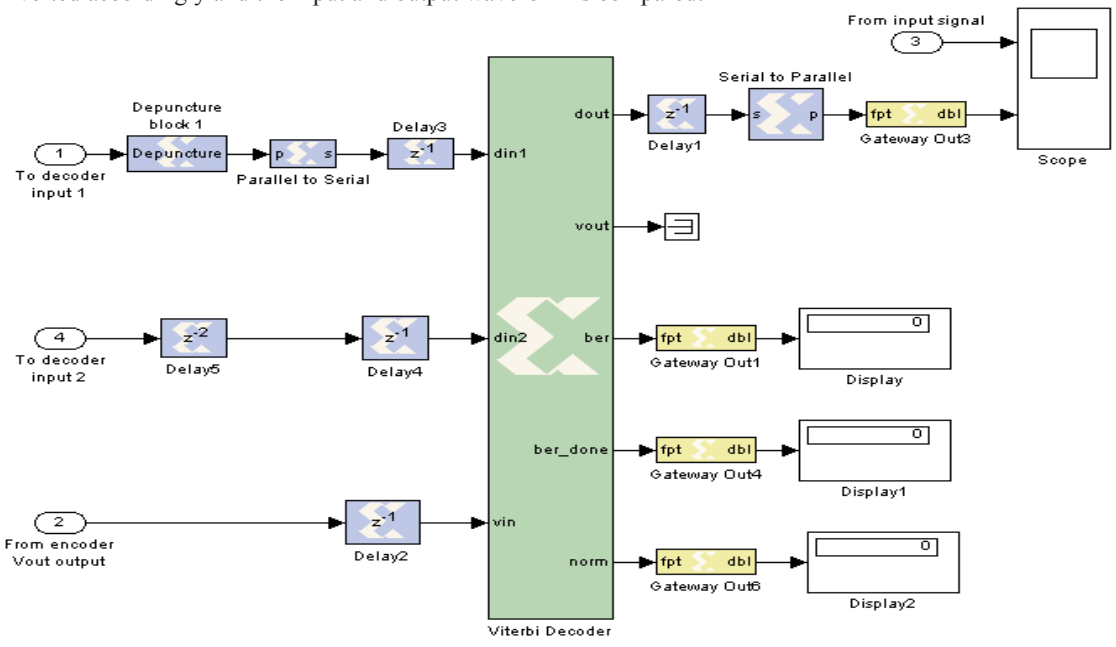

Fig.18 Depicturing and decoding model using System Generator blocks

6.2 Simulation Results

The constellation diagram and eye diagrams reveal the modulation characteristics of the signal and help to depict the impact of impairments, such as pulse shaping or channel distortions. They are commonly used to evaluate the overall performance of the digital communication systems. Since the channel used is AWGN, the extent to which the noise has affected the modulated signal can be seen from constellation and eye diagrams.

6.3 Constellation Diagrams

If the transmission error is less and the receiver output is more accurate. This is clearly reflected on Figure 20 compared to Figure 19. And in constellation diagram for BPSK shown in Figure 21. Recall that the power spectral density (PSD) of a QPSK signal has a null-to-null bandwidth that is equal to the bit rate, which is half that of a BPSK signal. Therefore, QPSK has twice the bandwidth efficiency of BPSK, since 2 bits are transmitted in a single modulation symbol instead of 1 bit for BPSK. Further, the bit error probability of QPSK is identical to BPSK, while twice as much data can be sent in the same bandwidth. Thus, when compared to BPSK, QPSK provides twice the spectral efficiency with exactly the same energy efficiency.

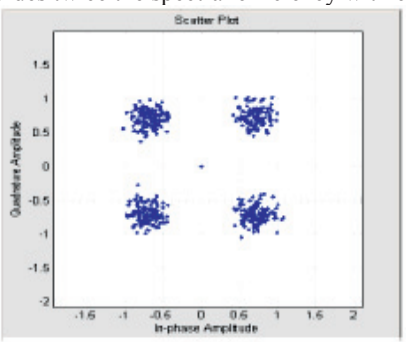

Fig.19 Constellation diagram with SNR=15dB

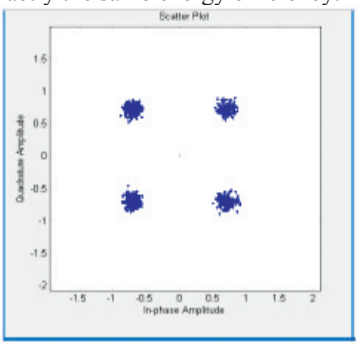

Fig.20 Constellation diagram with SNR =20dB

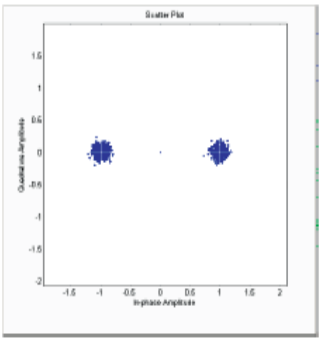

Fig.21 Constellation diagram SNR=20dB

6.4 Eye Diagrams

The measure of distortion, timing jitters and noise margin can be found from the eye diagrams. Figure 22 and 23 shows the eye diagram of the modulated signal for SNR of 15dB and 20dB.Comparing Figures 22 and 23 with noise in channel, it can be seen that the distortion in Figure 23 is 0.25 when compared with the distortion in Figure 22 which is 0.5. Similarly, the timing jitter is 0.1 for SNR = 20 dB and 0.2 for SNR = 15 dB. The other parameter noise margin is 0.65 and 0.5 for SNR = 30 dB and 15 dB respectively.

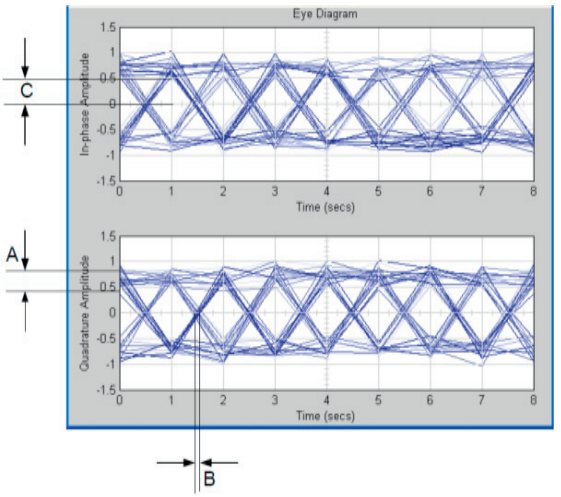

Fig.22 Eye diagram for QPSK modulated signal with SNR = 15 dB.

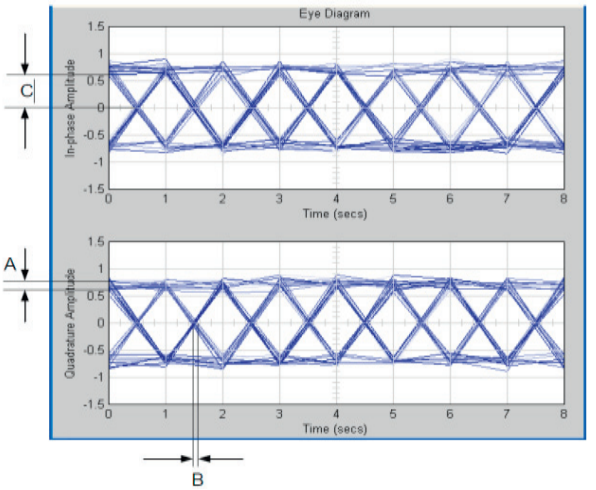

Fig.23 Eye diagram for QPSK modulated signal with SNR = 20 dB.

6.5 Output Waveform

Another indicator of performance is the observation of the output waveform compared to the input waveform. Figure 24 shows the input signal to the transmitter and the output signal of the receiver when both are synchronized and showing that the received signal is demodulated and decoded without much of error.

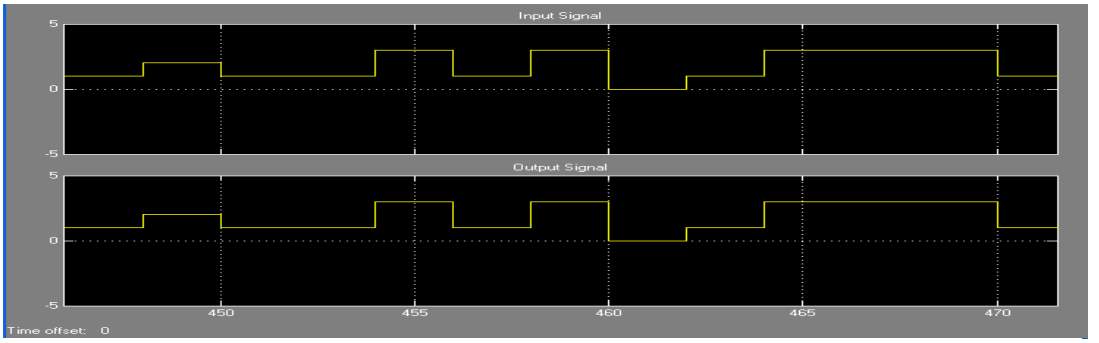

Fig.24 Tx/Rx Response

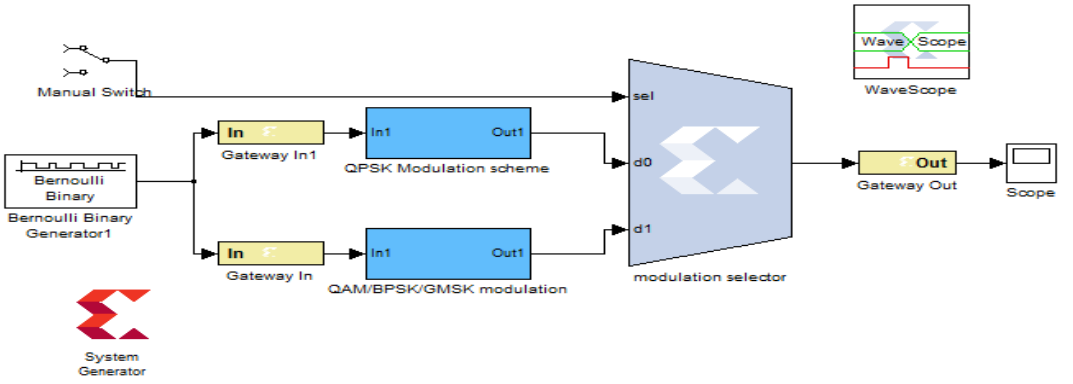

Fig.25 Baseband modulator on FPGA

7. FPGA implementation with Hardware – Software Co-simulation

The proposed reconfigurable hardware architecture for multidata transmission over FPGA by using digital modulation scheme identification has been shown in Figure 25.In the proposed The simulink-system generator software model, the system generator model is converted into HDL code for configuring on FPGA to function as multidata transmitter by using Simulink HDL Coder v1.4 and when we run the simulation, bit stream of data will be generated and will be dumped into the FPGA, In this work, Xilinx Spartan 500S4fg320 FPGA device has been used to implement the proposed design. The host system is equipped with each of the modulation type parameters and it keeps the state of the FPGA configuration to the current demodulation position. Whenever the change in modulation scheme occurs, the FPGA is reconfigured by downloading the appropriate bit stream file from the ROM based on manual selection.

8. Conclusion

The novelty of this work is to design an event announcement system for a campus and the device, that can be reconfigured, which consisting of single transmitter module which is implemented based on the SDR principle and Multiple receiver based announcement systems, which are located at various locations of the campus. The desired data transmitter and receiver are designed using BPSK and QPSK modulation and demodulations, implemented on FPGA. The data controlling for Tx and Rx is achieved using modulation schemes. The proposed design is prototyped on FPGA. Future scope of this project is to make it board level system design into commercially implementation, so that rapid access to the technology will become available at a simple software download.

References

[1]. Shyamnath Gollakota and Dina Katabi, 2008, "ZigZag decoding: Combating hidden terminals in wireless networks", in ACM-SIGCOMM'08, Seattle, WA. Vol. 38, pp. 159-170.

[2]. Mythili Vutukuru, Kyle Jamieson, and Hari Balakrishnan,NSDI'08, "Harnessing Exposed Terminals in Wireless Networks". Vol.37 ,p.59-72, April 16-18, 2008

[3]. Kyle Jamieson, 2008. "The SoftPHYAbstraction: from Packets to Symbols in Wireless Network Design. PhD thesis", MIT, Cambridge, MA.

[4]. Thomas Moscibroda, Ranveer Chandra, Yunnan Wu, Sudipta Sengupta, and Paramvir Bahl, 2008, "Load-aware spectrum distribution in wireless LANs", In IEEE ICNP. pp.137,146.

[5]. Sachin Katti, Shyamnath Gollakota, and Dina Katabi, 2007, "Embracing wireless interference: analog network coding", in SIGCOMM'07, Kyoto, Japan. Vol. 37, no. 4, pp 397-408.

[6]. Sachin Katti, Dina Katabi, Hari Balakrishnan, and Muriel Medard, 2008, "Symbol-Level Network Coding for Wireless Mesh Networks", in SIGCOMM'08, Seattle, WA. Vol 38, no. 4, pp 401-412.

[7]. Sachin Katti, Hariharan Rahul, Wenjun Hu, Dina Katabi, Muriel Medard, and Jon Crowcroft, 2006, "XORs in the air: practical wireless network coding", in SIGCOMM '06: Proceedings of the ACM SIGCOMM 2006 Conference on Data communication. Vol. 36 ,no 4, pp 243-254.

[8]. Hariharan Rahul, Nate Kushman, Dina Katabi, Charles Sodini, and Farinaz Edalat, 2008, "Learning to share: narrowband-friendly wideband networks", in SIG-COMM'08, Seattle, WA, USA. Vol 38, no 4, pp 147-158.

[9]. Naveen Kumar Santhapuri, Justin Manweiler, Souvik Sen, Roy Choudhury, Srihari Nelakuditi, and Kamesh Munagala,2008, "Message in message (MIM): A case for shuing transmissions in wireless networks", in Hotnets-VII, vol.44,pp-127-134, Calgary, Canada.

[10]. Andrew J. Viterbi, 1967, "Error bounds for convolutional codes and an asymptotically optimum decoding algorithm", in IEEE Trans. Info. Theory. Vol.13 , no 2 ,pp 260 – 269.

[11]. George Nychis, Thibaud Hottelier, Zhuochen Yang, Srinivasan Seshan, and Peter Steenkiste, 2009, "Enabling MAC Protocol Implementations on Software-defined Radios", in NSDI'09, Boston, MA. Vol.38,pp.91 -105.

[12]. Engling Yeo Stephanie Augsburger, Wm. Rhett Davis, and Borivoje Nikolic, 2002, "500Mb/s Soft Output Viterbi Decoder", ESSCIRC'02. Vol.37,pp.523-526.

[13]. G. Jo, M. Sheen, S. Lee and K. Cho, 2005, "A DSP-based reconfigurable SDR platform for 3G systems," *IEICE Trans. on Commun.*, Vol.E88-B No.2 pp. 678-683.

[14]. A. Haghighat, 2002, "A review on essentials and technical challenges of software defined radio," *Proceedings of MILCOM*, vol. 1, pp. 377-382.

[15]. K. Moessner, D. Bourse, D. Greifendorf and J. Stammen, 2003, "Software radio and reconfiguration management," *Computer Commun.*, vol. 26, issue 1 , pp. 26-35.

[16]. K. Compton and S. Hauck, 2002, "Reconfigurable computing: A survey of systems andsoftware," *ACM Computing Surveys*, vol. 34, issue 2, pp. 171-210.